Learning Multiple Visual Tasks while Discovering their Structure

Carlo Ciliberto * Lorenzo Rosasco * † Silvia Villa *

Abstract

Multi-task learning is a natural approach for computer vision applications that require the simultaneous solution of several distinct but related problems, e.g. object detection, classification, tracking of multiple agents, or denoising, to name a few. The key idea is that exploring task relatedness (structure) can lead to improved performances.

In this paper, we propose and study a novel sparse, non-parametric approach exploiting the theory of Reproducing Kernel Hilbert Spaces for vector-valued functions. We develop a suitable regularization framework which can be formulated as a convex optimization problem, and is provably solvable using an alternating minimization approach. Empirical tests show that the proposed method compares favorably to state of the art techniques and further allows to recover interpretable structures, a problem of interest in its own right.

1 Introduction

Several problems in computer vision and image processing, such as object detection/classification, image denoising, inpainting etc., require solving multiple learning tasks at the same time. In such settings a natural question is to ask whether it could be beneficial to solve all the tasks jointly, rather than separately. This idea is at the basis of the field of multi-task learning, where the joint solution of different problems has the potential to exploit tasks relatedness (structure) to improve learning. Indeed, when knowledge about task relatedness is available, it can be profitably incorporated in multi-task learning approaches for example by designing suitable embedding/coding schemes, kernels or regularizers, see [20, 10, 1, 11, 19].

The more interesting case, when knowledge about the tasks structure is not known a priori, has been the subject of recent studies. Largely influenced by the success of sparsity based methods, a common approach has been that of considering linear models for each task coupled with suitable parameterization/penalization enforcing task relatedness, for example encouraging the selection of features simultaneously important for all tasks [2] or for specific subgroups of related tasks [13, 14, 29, 15, 12, 16]. Other

*Laboratory for Computational and Statistical Learning, Istituto Italiano di Tecnologia, Via Morego, 30, 16100, Genova, Italy, (cciliber@mit.edu)

†DIBRIS, Università di Genova, Via Dodecaneso, 35, 16146, Genova, Italy, (lrosasco@mit.edu)

linear methods adopt hierarchical priors or greedy approaches to recover the taxonomy of tasks [22, 24]. A different line of research has been devoted to the development of non-linear/non-parametric approaches using kernel methods – either from a Gaussian process [1, 29] or a regularization perspective [1, 8].

This paper follows this last line of research, tackling in particular two issues only partially addressed in previous works. The first is the development of a regularization framework to learn and exploit the tasks structure, which is not only important for prediction, but also for interpretation. Towards this end, we propose and study a family of matrix-valued reproducing kernels, parametrized so to enforce sparse relations among tasks. A novel algorithm dubbed Sparse Kernel MTL is then proposed considering a Tikhonov regularization approach. The second contribution is to provide a sound computational framework to solve the corresponding minimization problem. While we follow a fairly standard alternating minimization approach, unlike most previous work we can exploit results in convex optimization to prove the convergence of the considered procedure. The latter has an interesting interpretation where supervised and unsupervised learning steps are alternated: first, given a structure, multiple tasks are learned jointly, then the structure is updated. We support the proposed method with an experimental analysis both on synthetic and real data, including classification and detection datasets. The obtained results show that Sparse Kernel MTL can achieve state of the art performances while unveiling the structure describing tasks relatedness.

The paper is organized as follows: in Sec. 2 we provide some background and notation in order to motivate and introduce the Sparse Kernel MTL model. In Sec. 3 we discuss an alternating minimization algorithm to provably solve the learning problem proposed. Finally, we discuss empirical evaluation in Sec. 4.

Notation. With $S^n_{++} \subset S^n_+ \subset S^n \subset \mathbb{R}^{n\times n}$ we denote respectively the space of positive definite, positive semidefinite (PSD) and symmetric $n \times n$ real-valued matrices. O^n denotes the space of orthonormal $n \times n$ matrices. For any $M \in \mathbb{R}^{n\times m}$, $M^\top$ denotes the transpose of M. For any PSD matrix $A \in S^n_+$, $A^\dagger \in S^n_+$ denotes the pseudoinverse of A. We denote by $I_n \in S^n_{++}$ the $n \times n$ identity matrix. We use the abbreviation l.s.c. to denote lower semi-continuous functions (i.e. functions with closed sub-level sets) [6].

2 Model

We formulate the problem of solving multiple learning tasks as that of learning a vector-valued function whose output components correspond to individual predictors. We consider the framework originally introduced in [20] where the well-known concept of Reproducing Kernel Hilbert Space is extended to spaces of vector-valued functions. In this setting the set of tasks relations has a natural characterization in terms of a positive semidefinite matrix. By imposing a sparse prior on this object we are able to formulate our model, Sparse Kernel MTL, as a kernel learning problem designed to recover the most relevant relations among the tasks.

In the following we review basic definitions and results from the theory of Reproducing Kernel Hilbert Spaces that will allow in Sec. 2.2 to motivate and introduce our learning framework. In Sec. 2.2.2 we briefly draw connections of our method to

previously prosed multi-task learning approaches.

2.1 Reproducing Kernel Hilbert Spaces for Vector-Valued Functions

We consider the problem of learning a function $f : \mathcal{X} \to \mathcal{Y}$ from a set of empirical observations $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y} \subseteq \mathbb{R}^T$. This setting includes learning problems such as vector-valued regression ($\mathcal{Y} = \mathbb{R}^T$), multi-label/detection for T tasks ($\mathcal{Y} = \{0, 1\}^T$) or also T-class classification (where we adopt the standard one-vs-all approach mapping the t-th class label to the t-th element e_t of the canonical basis in $\mathbb{R}^T$). Following the work of Micchelli and Pontil [20], we adopt a Tikhonov regularization approach in the setting of Reproducing Kernel Hilbert Spaces for vector-valued functions (RKHSvv). RKHSvv are the generalization of the well-known RKHS to the vector-valued setting and maintain most of the properties of their scalar counterpart. In particular, similarly to standard RKHS, RKHSvv are uniquely characterized by an operator-valued kernel:

Definition 2.1. *Let $\mathcal{X}$ be a set and $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ be a Hilbert space of functions from $\mathcal{X}$ to $\mathbb{R}^T$. A symmetric, positive definite, matrix valued function $\Gamma : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{T \times T}$ is called a reproducing kernel for $\mathcal{H}$ if for all $x \in \mathcal{X}, c \in \mathbb{R}^T$ and $f \in \mathcal{H}$ we have that $\Gamma(x, \cdot)c \in \mathcal{H}$ and the following reproducing property holds: $\langle f(x), c \rangle_{\mathbb{R}^T} = \langle f, \Gamma(x, \cdot)c \rangle_{\mathcal{H}}$.*

Analogously to the scalar setting, a Representer theorem holds, stating that the solution to the regularized learning problem

$$\underset{f \in \mathcal{H}}{\text{minimize}} \ \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}}^2 \tag{1}$$

is of the form $f(\cdot) = \sum_{i=1}^n \Gamma(\cdot, x_i)c_i$ with $c_i \in \mathbb{R}^T$, Γ the matrix-valued kernel associated to the RKHSvv $\mathcal{H}$ and $V : \mathcal{Y} \times \mathbb{R}^T \to \mathbb{R}_+$ a loss function (e.g. least squares, hinge, logistic, etc.) which we assume to be convex. We point out that the setting above can also account for the case where not all task outputs $y_i = (y_{i1}, \ldots, y_{iT})^\top$ associated to a given input x_i are available in training. Such situation would arise for instance in multi-detection problems in which supervision (e.g. presence/absence of an object class in the image) is provided only for a few tasks at the time.

2.1.1 Separable Kernels

Depending on the choice of operator-valued kernel Γ, different structures can be enforced among the tasks; this effect can be observed by restricting ourselves to the family of *separable* kernels. Separable kernels are matrix-valued functions of the form $\Gamma(x, x') = k(x, x')A$, where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a scalar reproducing kernel and $A \in S_+^T$ a $T \times T$ positive semidefinite (PSD) matrix. Intuitively, the scalar kernel characterizes the individual tasks functions, while the matrix A describes how they are related. Indeed, from the Representer theorem we have that solutions of problem (1) are of the form $f(\cdot) = \sum_{i=1}^n k(\cdot, x_i)Ac_i$ with the t-th task being $f_t(\cdot) =$

$\sum_{i=1}^{n} k(\cdot, x_i)\langle A_t, c_t\rangle_{\mathbb{R}^T}$, a scalar function in the RKHS $\mathcal{H}_k$ associated to kernel k. As shown in [10], in this case the squared norm associated to the separable kernel kA in the RKHSvv $\mathcal{H}$, can be written as

$$\|f\|_{\mathcal{H}}^2 = \sum_{t,s}^{T} A_{ts}^{\dagger}\langle f_t, f_s\rangle_{\mathcal{H}_k} \tag{2}$$

with $A_{ts}^{\dagger}$ the (t, s)-th entry of A's pseudo-inverse.

Eq. (2) shows how A can model the structural relations among tasks by directly coupling predictors: for instance, by setting $A^{\dagger} = I_T + \gamma(\mathbf{1}\mathbf{1}^{\top})/T$, with $\mathbf{1} \in \mathbb{R}^T$ the vector of all 1s, we have that the parameter γ controls the variance $\sum_{t=1}^{T} \|\bar{f} - f_t\|_{\mathcal{H}_k}^2$ of the tasks with respect to their mean $\bar{f} = \frac{1}{T}\sum_{t=1}^{T} f_t$. If we have access to some notion of similarity among tasks in the form of a graph with adjacency matrix $W \in S^T$, we can consider the regularizer $\sum_{t,s=1}^{T} W_{ts}\|f_t - f_s\|_{\mathcal{H}_k}^2 + \gamma\sum_{t}^{T}\|f_t\|_{\mathcal{H}_k}^2$ which corresponds to setting $A^{\dagger} = L + \gamma I_T$ with L the graph Laplacian induced by W. We refer the reader to [10] for more examples of possible choices for A when the tasks structure is known.

2.2 Sparse Kernel Multi Task Learning

When a-priori knowledge of the problem structure is not available, it is desirable to learn the tasks relations directly from the data. In light of the observations of Sec. 2.1.1, a viable approach is to parametrize the RKHSvv $\mathcal{H}$ in problem (1) with the associated separable kernel kA and to optimize jointly with respect to both $f \in \mathcal{H}$ and $A \in S_+^T$. In the following we show how this problem corresponds to that of identifying a set of latent tasks and to combine them in order to form the individual predictors. By enforcing a sparsity prior on the set of such possible combinations, we then propose the Sparse Kernel MTL model, which is designed to recover only the most relevant tasks relations. In Sec. 2.2.2 we discuss, from a modeling perspective, how our framework is related to the current multi-task learning literature.

2.2.1 Recovering the Most Relevant Relations

From the Representer theorem introduced in Sec. 2.1 we know that a candidate solution $f : \mathcal{X} \to \mathbb{R}^T$ to problem (1) can be parametrized in terms of the maps $k(\cdot, x_i)$, by a structure matrix $A \in S_+^T$ and a set of coefficient vectors $c_1, \ldots, c_n \in \mathbb{R}^T$ such that $f(\cdot) = \sum_{i=1}^{n} k(\cdot, x_i)Ac_i$. If now we consider the t-th component of f (i.e. the predictor of the t-th task), we have that

$$f_t(\cdot) = \sum_{i=1}^{n} k(\cdot, x_i)\langle A_t, c_i\rangle_{\mathbb{R}^T} = \sum_{s=1}^{T} A_{ts} g_s(\cdot) \tag{3}$$

where we set $g_s(\cdot) = \sum_{i=1}^{n} k(\cdot, x_i)c_{is} \in \mathcal{H}_k$ for $s \in \{1, \ldots, T\}$ and $c_{is} \in \mathbb{R}$ the s-th component of c_i. Eq. (3) provides further understanding on how A can enforce/describe the tasks relations: The g_s can be interpreted as elements in a dictionary and each f_t

factorizes as their linear combination. Therefore, any two predictors f_t and $f_{t'}$ are implicitly coupled by the subset of common g_s.

We consider the setting where the tasks structure is unknown and we aim to recover it from the available data in the form of a structure matrix A. Following a denoising/feature selection argument, our approach consists in imposing a sparsity penalty on the set of possible tasks structures, requiring each predictor f_t to be described by a small subset of g_s. Indeed, by requiring most of A's entries to be equal to zero, we implicitly enforce the system to recover only the most relevant tasks relations. The benefits of this approach are two-fold: on the one hand it is less sensitive to spurious statistically non-significant tasks-correlations that could for instance arise when few training examples are available. On the other hand it provides us with interpretable tasks structures, which is a problem of interest in its own right and relevant, for example, in cognitive science [17].

Following the de-facto standard choice of ℓ_1-norm regularization to impose sparsity in convex settings, the *Sparse Kernel MTL* problem can be formulated as

$$\underset{f\in\mathcal{H},A\in S_{++}^T}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} V(y_i, f(x_i)) + \\ \lambda(\|f\|_{\mathcal{H}}^2 + \epsilon\,\mathrm{tr}(A^{-1}) + \mu\,\mathrm{tr}(A) + (1-\mu)\,\|A\|_{\ell_1}) \quad (4)$$

where $\|A\|_{\ell_1} = \sum_{t,s}|A_{ts}|$, $V : \mathcal{Y}\times\mathbb{R}^T \to \mathbb{R}_+$ is a loss function and $\lambda > 0$, $\epsilon > 0$, and $\mu \in [0,1]$ regularization parameters. Here $\mu \in [0,1]$ regulates the amount of desired entry-wise sparsity of A with respect to the low-rank prior $tr(A)$ (indeed notice that for $\mu = 1$ we recover the low-rank inducing framework of [2, 28]). This prior was empirically observed (see [2, 28]) to indeed encourage information transfer across tasks; the sparsity term can therefore be interpreted as enforcing such transfer to occur only between tasks that are strongly correlated. Finally the term $\epsilon\,\mathrm{tr}(A^{-1})$ ensures the existence of a unique solution (making the problem strictly convex), and can be interpreted as a preconditioning of the problem (see Sec. 3.2).

Notice that the term $\|f\|_{\mathcal{H}}^2$ depends on both f and A (see Eq. 2), thus making problem (4) non-separable in the two variables. However, it can be shown that the objective functional is jointly convex in f and A (we refer the reader to the Appendix for a proof of convexity, which extends results in [2] to our setting). This will allow in Sec. 3 to derive an optimization strategy that is guaranteed to converge to a global solution.

2.2.2 Previous Work on Learning the Relations among Tasks

Several methods designed to recover the tasks relations from the data can be formulated using our notation as joint learning problems in f and A. Depending on the expected/desired tasks-structure a set of constraints $\mathcal{A} \subseteq S_{++}^T$ can be imposed on A when solving a joint problem as in (4):

- **Multi-task Relation Learning** [28]. In [28], the relaxation $\mathcal{A} = \{A|\,\mathrm{tr}(A) \leq 1\}$ of the low-rank constraint is imposed, enforcing the tasks f_t to span a low-dimensional subspace in $\mathcal{H}_k$. This method can be shown to be approximately equivalent to [2].

- **Output Kernel Learning** [8]. Rather than imposing a hard constraint, the authors penalize the structure matrix A with the squared Frobenius norm $\|A\|_F^2$.
- **Cluster Multi-task Learning** [13]. Assuming tasks to be organized into distinct clusters, in [13] a learning scheme to recover such structure is proposed, which consists of imposing a suitable set of spectral constraints $\mathcal{A}$ on A. We refer the reader to the supplementary material for further details.
- **Learning Graph Relations** [3]. Following the interpretation in [10] reviewed in Sec. 2.1.1 of imposing similarity relations among tasks in the form of a graph, in [3] the authors propose a setting where a (relaxed) Graph Laplacian constraint is imposed on A.

3 Optimization

Due to the clear block variable structure of Eq. (4) with respect to f and A, we propose an alternating minimization approach (see Alg. 1) to iteratively solve the Sparse Kernel MTL problem by keeping fixed one variable at the time. This choice is motivated by the fact that for a fixed A, problem (4) reduces to the standard multi-task learning problem (1), for which several well-established optimization strategies have already been considered [1, 20, 10, 21]. The alternating minimization procedure can be interpreted as iterating between steps of supervised learning (finding the f that best fits the input-output training observations) and unsupervised learning (finding the best A describing the tasks structure, which does not involve the output data).

3.1 Solving w.r.t. f (Supervised Step)

Let $A \in S_{++}^T$ be a fixed structure matrix. From the Representer theorem (see Sec. 2.1) we know that the solution of problem (1) is of the form $f(\cdot) = \sum_{i=1}^n k(\cdot, x_i) A c_i$ with $c_i \in \mathbb{R}^T$. Depending on the specific loss V, different methods can be employed to find such coefficients c_i. In particular, for the least-square loss a closed form solution can be derived by taking the coefficient vector $c = (c_1^\top, \dots, c_n^\top)^\top \in \mathbb{R}^{nT}$ to be [1]:

$$c = (A \otimes K + \lambda I_{nT})^{-1} y \tag{5}$$

where $K \in S_+^n$ is the empirical kernel matrix associated to k the scalar kernel, $y \in \mathbb{R}^{nT}$ is the vector concatenating the training outputs $y_1, \dots, y_n \in \mathbb{R}^T$ and $\otimes$ denotes the Kronecker product. A faster and more compact solution was proposed in [21] by adopting Sylvester's method.

3.2 Solving w.r.t the Tasks Structure (Unsupervised Step)

Let f be known in terms of its coefficents $c_1, \dots, c_n \in \mathbb{R}^T$. Our goal is to find the structure matrix $A \in S_{++}^T$ that minimizes problem (4). Notice that each task f_t can be written as $f_t(\cdot) = \sum_{i=1}^n k(\cdot, x_i) \langle A_t, c_i \rangle_{\mathbb{R}^T} = \sum_{i=1}^n k(\cdot, x_i) b_{i,t}$ with $b_{i,t} = \langle A_t, c_i \rangle_{\mathbb{R}^T}$.

Algorithm 1 ALTERNATING MINIMIZATION

Input: K empirical kernel matrix, y training outputs, δ tolerance, V loss, λ, μ, ϵ hyperparameters, S objective functional of problem (4).
Initialize: $f_0 = 0$, $A_0 = I_T$ and $i = 0$
repeat
  $f_{i+1} \leftarrow$ SUPERVISEDSTEP (V, K, y, A_i, λ)
  $A_{i+1} \leftarrow$ SPARSEKERNELMTL$(K, f_{i+1}, \mu, \epsilon)$
  $i \leftarrow i + 1$
until $|S(f_{i+1}, A_{i+1}) - S(f_i, A_i)| < \delta$

Therefore, from eq. (2) we have

$$\|f\|_{\mathcal{H}}^2 = \sum_{t,s}^{T} A_{ts}^{-1} \langle f_t, f_s \rangle_{\mathcal{H}_k} = \sum_{t,s}^{T} \sum_{i,j} A_{ts}^{-1} k(x_i, x_j) b_{it} b_{js} \tag{6}$$

where we have used the reproducing property of $\mathcal{H}_k$ for the last equality. Eq. (6) allows to write the norm induced by the separable kernel kA in the more compact matrix notation $\|f\|_{\mathcal{H}}^2 = \operatorname{tr}(B^\top K B A^{-1})$, where $B \in \mathbb{R}^{n \times T}$ is the matrix with (i,t)-th element $B_{it} = b_{it}$.

Under this new notation, problem (4) with fixed f becomes

$$\min_{A \in S_{++}^T} \operatorname{tr}(A^{-1}(B^\top K B + \epsilon I_T)) + \mu \operatorname{tr}(A) + (1-\mu) \|A\|_{\ell_1} \tag{7}$$

from which we can clearly see the effect of ϵ as a preconditioning term for the tasks covariance matrix $B^\top K B$.

By employing recent results from the non-smooth convex optimization literature, in the following we will describe an algorithm to optimize the Sparse Kernel MTL problem.

3.2.1 Primal-dual Splitting Algorithm

First order proximal splitting algorithms have been successfully applied to solve convex composite optimization problems, that can be written as the sum of a smooth component with nonsmooth ones [4]. They proceed by splitting, i.e. by activating each term appearing in the sum individually. The iteration usually consists of a gradient descent-like step determined by the smooth component, and various proximal steps induced by the nonsmooth terms [4]. In the following we will describe one of such methods, derived in [26, 7], to solve the Sparse Kernel MTL problem in eq. (7). The proposed method is primal-dual, in the sense that it also provides an additional dual sequence solving the associated dual optimization problem. We will rely on the sum structure of the objective function, that can be written as $G(\cdot) + H_1(\cdot) + H_2(L(\cdot))$, with $G(A) = \lambda\mu \operatorname{tr}(A)$, $H_1(A) = \lambda(1-\mu)\|A\|_{\ell_1}$ and $H_2(A) = \lambda\epsilon \operatorname{tr}(A^{-1}) + i_{S_{++}^T}(A)$, where $i_{S_{++}^T}$ is the indicator function of a S_{++}^T (0 on the set $+\infty$ outside) and enforces the hard constraint $A \in S_{++}^T$. L is a linear operator defined as $L(A) = MAM$, where

we have set $M = (B^\top KB + \epsilon I_T)^{-1/2}$. We recall here that a square root of a PSD matrix $P \in S_+^T$ is a PSD matrix $M \in S_+^T$ such that $P = MM$. Note that G is smooth with Lipschitz continuous gradient, L is a linear operator and both H_1 and H_2 are functions for which the proximal operator can be computed in closed form. We recall that the proximity operator at a point $y \in \mathbb{R}^m$ of a proper, convex and l.s.c. function $H : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$, is defined as

$$\mathrm{prox}_H(y) = \operatorname*{argmin}_{x \in \mathbb{R}^m} \Big\{ H(x) + \frac{1}{2}\|x - y\|^2 \Big\}. \tag{8}$$

It is well known that for any $\eta > 0$, the proximal map of the ℓ_1 norm $\eta\|\cdot\|_{\ell_1}$ is the so-called *soft-thresholding* operator $S_\eta(\cdot)$, which can be computed in closed form. The following result provides an explicit closed-form solution also for the proximal map of H_2.

Proposition 3.1. *Let $Z \in S^T$ with eigendecomposition $Z = U\Sigma U^\top$ with $U \in O^T$ orthonormal matrix and $\Sigma \in S^T$ diagonal. Then*

$$\mathrm{prox}_{H_2}(Z) = \operatorname*{argmin}_{A \in S_{++}^T} \Big\{ \mathrm{tr}(A^{-1}) + \frac{1}{2}\|A - Z\|_F^2 \Big\}. \tag{9}$$

can be computed in closed form as $\mathrm{prox}_{H_2}(Z) = U\Lambda U^\top$ with $\Lambda \in S_{++}^T$ diagonal matrix with Λ_{tt} the only positive root of the polynomial $p(\lambda) = \lambda^3 - \lambda^2\Sigma_{tt} - 1$ with $\lambda \in \mathbb{R}$.

Proof. Note that H_2 is convex and lsc. Therefore the proximity operator is well-defined and the functional in (9) has a unique minimizer. Its gradient is $-A^{-2} + A - Z$, therefore, the first order condition for a matrix A to be a minimizer is

$$A^3 - A^2 Z - I_T = 0 \tag{10}$$

We show that it is possible to find $\Lambda \in S_{++}^T$ diagonal such that $A_* = U\Lambda U^\top$ solves eq. (10). Indeed, for A with same set of eigenvectors U as Z, we have that eq. (10) becomes $U(\Lambda^3 - \Lambda^2\Sigma - I_T)U^\top = 0$, which is equivalent to the set of T scalar equations $\lambda^3 - \lambda^2\Sigma_{tt} - 1 = 0$ for $t \in \{1, \dots, T\}$ and $\lambda \in \mathbb{R}$. Descartes rule of sign [23] assures that for any $\Sigma_{tt} \in \mathbb{R}$ each of these polynomials has exactly one positive root, which can be clearly computed in closed form. □

We have the following result as an immediate consequence.

Theorem 3.2 (Convergence of Sparse Kernel MTL, [26, 7])**.** *Let k be a scalar kernel over a space $\mathcal{X}$, $x_1, \dots, x_n \in \mathcal{X}$ a set of points and $f : \mathcal{X} \to \mathbb{R}^T$ a function characterized by a set of coefficients $b_1, \dots, b_n \in \mathbb{R}^T$ so that $f(\cdot) = \sum_{i=1}^n k(\cdot, x_i)b_i$. Set $K \in S_+^n$ to be the empirical kernel matrix associated to k and the points $\{x_i\}_{i=1}^n$ and $B \in \mathbb{R}^{n \times T}$ the matrix whose i-th row corresponds to the (transposed) coefficient vector b_i.*

Then, any sequence of matrices A_t produced by Algorithm (2) *converges to a global minimizer of the Sparse Kernel MTL problem* (4) *(or, equivalently, to* (7)*) for fixed f. Furthermore, the sequence D_t converges to a solution of the dual problem of* (7)*.*

Algorithm 2 SPARSE KERNEL MTL

Input: $K \in S_+^n$, $B \in \mathbb{R}^{n \times T}$, δ tolerance, $0 \leq \mu \leq 1, \epsilon > 0$ hyperparameter.
Initialize: $A_0, D_0 \in S_{++}^T$, $M = (B^\top K B + \epsilon I_T)^{-1/2}$, $\sigma = \|M\|^2$ squared maximum eigenvalue of M. $i = 0$
repeat
$A_{i+1} \leftarrow \operatorname{prox}_{\frac{1-\mu}{\sigma}\|\cdot\|_{\ell_1}}(A_i - \frac{1}{\sigma}(\mu I_T + M D_i M))$
$P \leftarrow D_i + \frac{1}{\sigma} M(2A_{i+1} - A_i)M$
$D_{i+1} \leftarrow P - \operatorname{prox}_{\sigma H_2}(\sigma P)$
$i \leftarrow i + 1$
until $\|A_{i+1} - A_i\|_F < \delta$ and $\|D_{i+1} - D_i\|_F < \delta$

3.3 Convergence of Alternating Minimization

We additionally exploit the sum structure and the regularity properties of the objective functional in (4) to prove convergence of the alternating minimization scheme to a global minimum. We rely on the results in [25]. In particular, the following result is a direct application of Theorem 4.1 in that paper.

Theorem 3.3. *Under the same assumptions as in Theorem 3.2, the sequence $(f_i, A_i)_{i \in \mathbb{N}}$ generated by Algorithm 1 is a minimizing sequence for Problem 4 and converges to its unique solution.*

Proof. Let S denote the objective function in (4). First note that the level sets of S are compact due to the presence of the term $\epsilon \operatorname{tr}(A^{-1}) + \mu \operatorname{tr}(A)$ and that S is continuous on each level set. Moreover, since S is regular at each point in the interior of the domain and is convex, [25, Theorem 4.1(c)] implies that each cluster point of $(f_i, A_i)_{i \in \mathbb{N}}$ is the unique minimizer of S. Then, the sequence itself is convergent and is minimizing by continuity. □

3.3.1 A Note on Computational Complexity & Times

Regarding the computational costs/number of iterations required for the convergence of the whole Alg. 1, up to our knowledge the only results available on rates for Alternating Minimization are in [5]. Unfortunately these results hold only for smooth settings. Notice however that each iteration of Alg 2 is of the order of $O(T^3)$, (the eigendecomposition of A being the most expensive operation) and its convergence rate is $O(1/k)$ with k equal to the number of iterations. Hence, Alg. 2 is not affected by the number n of training samples. On the contrary, the supervised step in Agl. 1 (e.g. RLS or SVM) typically requires the inversion of the kernel matrix K (or some approximation of its inverse) whose complexity heavily depends on n (order of $O(n^3)$ for inversion). Furthermore, the product $BKB^\top$ costs $O(n^2 T)$ which, since $n >> T$, is more expensive than Alg. 1. Thus, with respect to n SKMTL scales exactly as methods such as [2,7,24].

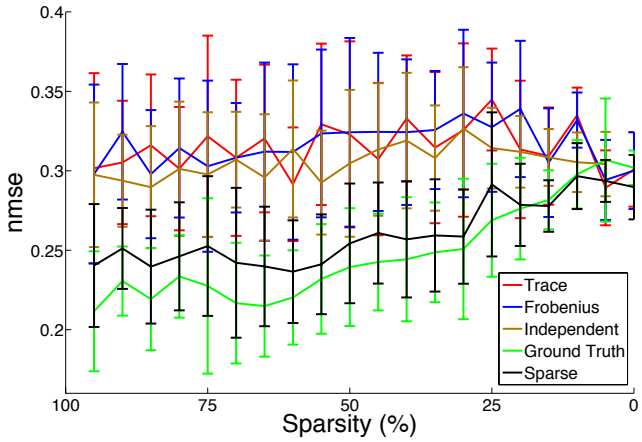

Figure 1: Generalization performance (nMSE and standard deviation) of different multi-task methods with respect to the sparsity of the task structure matrix.

4 Empirical Analysis

We report the empirical evaluation of SKMTL on artificial and real datasets. We have conducted experiments on both artificially generated and real dataset to assess the capabilities of the proposed Sparse Kernel MTL method to recover the most relevant relations among tasks and exploit such knowledge to improve the prediction performance.

4.1 Synthetic Data

We considered an artificial setting that allows us to control the tasks structure and in particular the actual sparsity of the tasks-relation matrix. We generated synthetic datasets of input-output pairs $(x, y) \in \mathbb{R}^d \times \mathbb{R}^T$ according to linear models of the form $y^\top = x^\top UA + \epsilon$ where $U \in \mathbb{R}^{d \times T}$ is a matrix with orthonormal columns, $A \in S_+^T$ is the task structure matrix and ϵ is zero-mean Gaussian noise with variance 0.1. The inputs $x \in \mathbb{R}^d$ were sampled according to a Gaussian distribution with zero mean and identity covariance matrix. We set the input space dimension $d = 100$ for our experiments.

In order to quantitatively control the sparsity level of the tasks-relation matrix, we randomly generated A so that the ratio between its support (i.e. the number of non-zero entries) and the total number of entries would vary between 0.1 (90% sparsity) and 1 (no sparsity). A Gaussian noise with zero mean and variance $1/10$ of the mean value of the non-zero entries in A was sampled to corrupt the structure matrix entries (hence, the model A was never "really" sparse). This was done to reproduce a more realistic scenario.

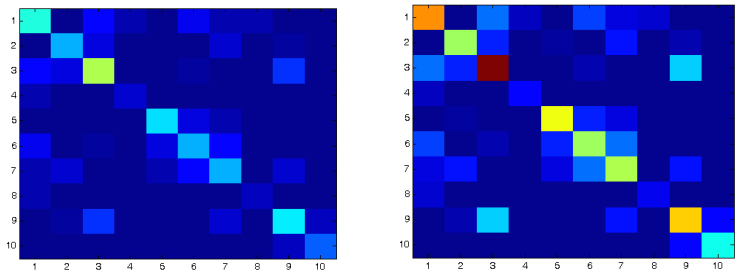

Figure 2: Structure matrix A. True (Left) and recovered by Sparse Kernel MTL (Right). We report the absolute value of the entries of the two matrices. The range of values goes from 0 (Blue) to 1 (Red)

We generated multiple models and corresponding datasets for different sparsity ratios and number of tasks T ranging from 5 to 20. For each dataset we generated respectively 50 samples for training and 100 for test. We performed multi-task regression using the following methods: single task learning (STL) as baseline, Multi-task Relation Learning [28] (MTRL), Output Kernel Learning [8] (OKL), our Sparse Kernel MTL (SKMTL) and a fixed task-structure multi-task regression algorithm solving problem (1) using the ground truth (GT) matrix A (after noise corruption) for regularization. We chose least-square loss and performed model selection with five-fold cross validation.

In Figure 1 we report the normalized mean squared error (nMSE) of tested method with respect to decreasing sparsity ratios. It can be noticed that knowledge of the true A (GT) is particularly beneficial when the tasks share few relations. This advantage tends to decrease as the tasks structure becomes less sparse. Interestingly, both the MTRL and OKL method do not provide any advantage with respect to the STL baseline since we did not design A to be low-rank (or have a fast eigenvalue decay). On the contrary, the SKMTL method provides a remarkable improvement over the STL baseline.

We point out that the large error bars in the plot are due to the high variability of the nMSE with respect to the different (random) linear models A and number of tasks T. The actual improvement of the SKMTL over the other methods is however significant.

The results above suggest that, as desired, our SKMTL method is actually recovering the most relevant relations among tasks. In support of this statement we report in Figure 2 an example of the true (uncorrupted) and recovered structure matrix A in the case of $T = 10$ and 50% sparsity. As can be noticed, while the actual values in the entries of the two matrices are not exactly the same, their supports almost coincide, showing that SKMTL was able to recover the correct tasks structure.

	Accuracy (%) per # tr. samples per class		
	50	100	150
STL	72.23	76.61	79.23
	±0.04	±0.02	±0.01
MTFL [2]	73.23	77.24	80.11
	±0.08	±0.05	±0.03
MTRL [28]	73.13	77.53	80.21
	±0.08	±0.04	±0.05
OKL [8]	72.25	77.06	80.03
	±0.03	±0.01	±0.01
SKMTL	**73.50**	**78.23**	**81.32**
	±0.11	±0.06	±0.08

Table 1: Classification results on the 15-scene dataset. Four multi-task methods and the single-task baseline are compared.

4.2 15-Scenes

We tested SKMTL in a multi-class classification scenario for visual scene categorization, the 15-scenes dataset[1]. The dataset contains images depicting natural or urban scenes that have been organized in 15 distinct groups and the goal is to assign each image to the correct scene category. It is natural to expect that categories will share similar visual features. Our aim was to investigate whether these relations would be recovered by the SKMTL method and result beneficial to the actual classification process.

We represented images in the dataset with LLC coding [27], trained multi-class classifiers on 50, 100 and 150 examples per class and tested them on 500 samples per class. We repeated these classification experiments 20 times to account for statistical variability.

In Table 1 we report the classification accuracy of the multi-class learning methods tested: STL (baseline), Multi-task Feature Learning (MTFL) [2], MTRL, OKL and our SKMTL. For all methods we used a linear kernel and least-squares loss as plug-in classifier. Model selection was performed by five-fold cross-validation.

As it can be noticed, the SKMTL consistently outperforms all other methods. A possible motivation for this behavior, similarly to the synthetic scenario, is that the algorithm is actually recovering the most relevant relations among tasks and using this information to improve prediction. In support of this interpretation, in Figure 3 we report the relations recovered by SKMTL in graph form. An edge between two scene categories t and s was drawn whenever the value of the corresponding entry A_{ts} of the recovered structure matrix was different from zero. Noticeably SKMTL seems to

[1] http://www-cvr.ai.uiuc.edu/ponce_grp/data/

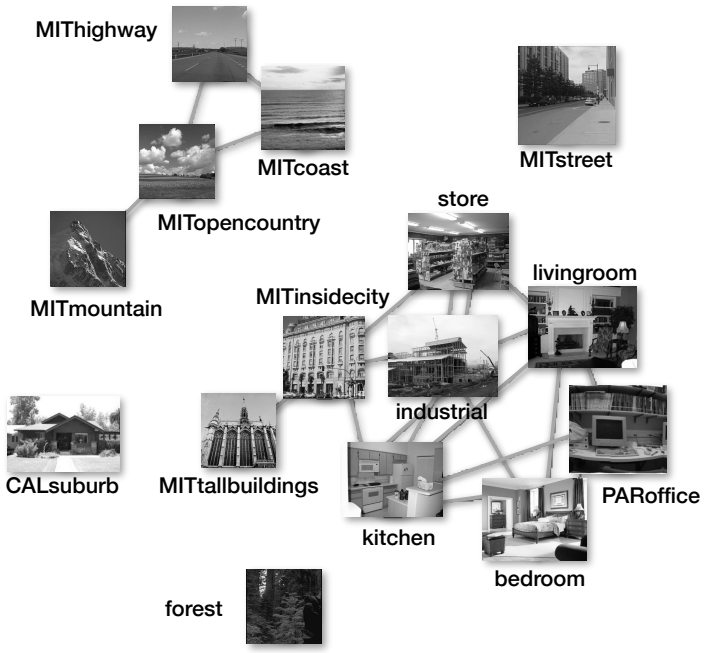

Figure 3: Tasks structure graph recovered by the Sparse Kernel MTL (SKMTL) proposed in this work on the 15-scenes dataset.

identify a clear group separation between natural and urban scenes. Furthermore, also within these two main clusters, not all tasks are connected: for instance office scenes are not related to scenes depicting the exterior of buildings or mountain scenes are not connected to images featuring mostly flat scenes such as highways or coastal regions.

4.3 Animals with Attributes

Animals with Attributes[2] (AwA) is a dataset designed to benchmark detection algorithms in computer vision. The dataset comprises 50 different animal classes each annotated with 85 binary labels denoting the presence/absence of different attributes. These attributes can be of different nature such as color (white, black, etc.), texture (stripes, dots), type of limbs (hands, flippers, etc.), diet and so on. The standard challenge is to perform attribute detection by training the system on a predefined set of 40 animal classes and testing on the remaining 10. In the following we will first discuss the performance of multi-task approaches in this setting and then investigate how the benefits of multi-task approaches can sometime be dulled by the so-called "negative transfer" and how our Sparse Kernel MTL method seems to be less sensitive to such an

[2] http://attributes.kyb.tuebingen.mpg.de/

	AUC (%) per #tr. samples per class		
	50	100	150
STL	57.26 ± 1.71	60.73 ± 1.12	64.37 ± 1.29
MTFL	58.11 ± 1.23	61.21 ± 1.14	64.22 ± 1.10
MTRL	58.24 ± 1.84	61.18 ± 1.23	64.56 ± 1.41
OKL	$\mathbf{58.81 \pm 1.18}$	62.07 ± 1.05	64.26 ± 1.18
SKMTL	58.63 ± 1.73	$\mathbf{63.21 \pm 1.43}$	64.51 ± 1.83

Table 2: Attribute detection results on the Animals with Attributes dataset.

issue. For the experiments described in the following we used the DECAF features [9] recently made available on the Animals With Attribute website.

4.3.1 Attribute Detection

We considered the multi-task problem of attribute detection which consists in 85 classification (binary) tasks. For each attribute, we randomly sampled 50, 100 and 150 examples for training, 500 for validation and 500 for test. Results were averaged over 10 trials. In Table 2 we report the Average Precision (area under the precision/recall curve) of the multi-task classifiers tested. As can be noticed for all multi-task approaches, the effect of sharing information across classifiers seems to have a remarkable impact when few training examples are available (the 50 or 100 columns in Table 2). As expected, such benefit decreases as the role of regularization becomes less crucial (150).

4.3.2 Attribute Prediction - Color Vs Limb Shape

Multi-task learning approaches ground on the assumption that tasks are strongly related one to the other and that such structure can be exploited to improve overall prediction. When this assumption doesn't hold, or holds only partially (e.g. only *some* tasks have common structure), such methods could even result disadvantageous ("negative transfer" [22]).

The AwA dataset offers the possibility to observe this effect since attributes are organized into multiple semantic groups [18, 14]. We focused on a smaller setting by selecting only two group of tasks, namely *color* and *limb shape*, and tested the effect of training multi-task methods jointly or independently across such two groups. For all the experiments we randomly sampled for each class 100 examples for training, 500 for validation and 500 for test, averaging the system performance over 10 trials. Table 3 reports the average precision separately for the color and limb shape groups.

Interestingly, methods relying on the assumption that all tasks share a common structure, such as MTFL, MTRL or OKL, experience a slight drop in performance when trained on all attribute detection tasks together (right columns) rather than separately (left column). On the contrary, SKMTL remains stable since it correctly separates the two groups.

	Area under PR Curve (%)			
	Independent		Joint	
	Color	**Limb**	**Color**	**Limb**
STL	74.33	68.13	74.33	68.15
	±0.81	±0.93	±0.81	±0.91
MTFL	75.21	69.41	74.98	69.71
	±0.73	±1.01	±1.18	±0.81
MTRL	75.17	69.18	74.92	69.73
	±0.53	±0.64	±0.78	±0.75
OKL	74.52	68.54	74.31	68.44
	±0.44	±0.61	±0.54	±0.22
SKMTL	75.14	69.21	75.23	69.57
	±0.97	±0.83	±0.77	±0.76

Table 3: Attribute detection on two subsets of AwA. Comparison between methods trained independently or jointly on the two sets show the effects of negative transfer.

5 Conclusions

We proposed a learning framework designed to solve multiple related tasks while simultaneously recovering their structure. We considered the setting of Reproducing Kernel Hilbert Spaces for vector-valued functions [20] and formulated the Sparse Kernel MTL as an output kernel learning problem where both a multi-task predictor and a matrix encoding the tasks relations are inferred from empirical data. We imposed a sparsity penalty on the set of possible relations among tasks in order to recover only those that are more relevant to the learning problem.

Adopting an alternating minimization strategy we were able to devise an optimization algorithm that provably converges to the global solution of the proposed learning problem. Empirical evaluation on both synthetic and real dataset confirmed the validity of the model proposed, which successfully recovered interpretable structures while at the same time outperformed previous methods.

Future research directions will focus mainly on modeling aspects: it will be interesting to investigate the possibility to combine our framework, which identifies sparse relations among the tasks, with recent multi-task linear models that take a different perspective and enforce tasks relations in the form of structured sparsity penalties on the feature space [14, 29].

References

[1] M. Álvarez, N. Lawrence, and L. Rosasco. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012. see also http://arxiv.org/abs/1106.6251.

[2] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73, 2008.

[3] A. Argyriou and Â. E. C. Paris. Learning the graph of relations among multiple tasks.

[4] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011. With a foreword by Hédy Attouch.

[5] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *Technion, Israel Institute of Technology, Haifa, Israel, Tech. Rep*, 2011.

[6] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] L. Condat. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.*, 158(2):460–479, 2013.

[8] F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. *International Conference on Machine Learning*, 2011.

[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.

[10] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. In *Journal of Machine Learning Research*, pages 615–637, 2005.

[11] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. *European Conference on Computer Vision*, 2010.

[12] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1761–1768. IEEE, 2011.

[13] L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: a convex formulation. *Advances in Neural Information Processing Systems*, 2008.

[14] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *CVPR*, 2014.

[15] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 521–528, 2011.

[16] A. Kumar and H. Daume III. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*, 2012.

[17] B. M. Lake and J. B. Tenenbaum. Discovering structure by learning sparse graphs. *Proceedings of the 32nd Cognitive Science Conference*, 2010.

[18] C. Lampert. Semantic attributes for object categorization (slides). *http://ist.ac.at/chl/talks/lampert-vrml2011b.pdf*, 2011.

[19] A. Lozano and V. Sindhwani. Block variable selection in multivariate regression and high-dimensional causal inference. *Advances in Neural Information Processing Systems*, 2011.

[20] C. A. Micchelli and M. Pontil. Kernels for multi-task learning. *Advances in Neural Information Processing Systems*, 2004.

[21] H. Q. Minh and V. Sindhwani. Vector-valued manifold regularization. *International Conference on Machine Learning*, 2011.

[22] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1481–1488. IEEE, 2011.

[23] D. J. Struik. A source book in mathematics 1200–1800. *Princeton University Press*, pages 89–93, 1986.

[24] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–762. IEEE, 2004.

[25] P. Tseng. Convergence of block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001.

[26] B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.*, 38(3):667–681, 2013.

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[28] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 733–742, Corvallis, Oregon, 2010. AUAI Press.

[29] W. Zhong and J. Kwok. Convex multitask learning with flexible task clusters. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 49–56, New York, NY, USA, July 2012. Omnipress.

6 Appendix

6.1 On the (joint) convexity of Sparse Kernel MTL

As stated in the paper, it can be shown that the Sparse Kernel MTL problem introduced in Eq. (4) is jointly convex in the two optimization variables f and A. The proof of this fact requires the introduction of functional analysis tools that are beyond the scope of this work. Indeed, according to equation (6) we have observed that it is possible to restrict the SKMTL problem to functions of the form $f(\cdot) = \sum_{i=1}^{n} k(\cdot, x_i)b_i$ with $b_i \in \mathbb{R}^T$. The following result proves the joint-convexity of Eq. (4) for this setting. It is an extension of similar results in [2, 28] and we give it here for completeness.

Proposition 6.1. *Let $V : \mathbb{R}^T \to \mathbb{R}^T \to \mathbb{R}_+$ be a convex loss function. Then the functional in problem* (4) *– restricted to functions f of the form $f(\cdot) = \sum_{i=1}^{n} k(\cdot, x_i)b_i$ with $b_i \in \mathbb{R}^T$ – is convex in both f and A.*

Proof. Notice that, the only term that requires some care is the component of the functional that is mixing f and A together, namely $\|f\|_{\mathcal{H}}$ (where the dependency to A is implicit in $\mathcal{H}$. Indeed, since V is chosen to be convex, the empirical risk term is clearly convex in f and does not depend on A, while all the remaining terms are – i.e. the $tr(A^{-1})$, $tr(A)$ and $\|A\|_{\ell_1}$ – penalize only the structure matrix A and are clearly convex with respect to it.

According to Eq. (6) $f(\cdot) = \sum_{i=1}^{n} k(\cdot, x_i)b_i$ and we have that $\|f\|_{\mathcal{H}}^2$ can be rewritten as $\|f\|_{\mathcal{H}}^2 = \mathrm{tr}(B^\top K B A^{-1}))$, with $K \in S_+^n$ the empirical kernel matrix and $B \in \mathbb{R}^{n \times T}$ the matrix whose rows correspond to $b_i^\top$. Let us now set $b = vec(B) \in \mathbb{R}^{nT}$ the vectorization of matrix B, obtained by concatenating the columns of B. Then we have that

$$tr(B^\top K B A^{-1}) = b^T (A^{-1} \otimes K) b. \tag{11}$$

In order to show that the function $Q(A, b) = b^T(A^{-1} \otimes K)b$ is jointly convex in b and A we will show that its epigraph is a convex set. To see this notice that

$$\begin{aligned} epi_Q &= \{(A, b, c) \in S_{++}^T \times \mathbb{R}^{nT} \times \mathbb{R} \mid c \geq w^\top (A^{-1} \otimes K) w\} \\ &= \{(A, b, c) \in S_{++}^T \times \mathbb{R}^{nT} \times \mathbb{R} \mid \begin{pmatrix} A \otimes K^\dagger & b \\ b^\top & c \end{pmatrix} \in S_+^{nt+1}\} \end{aligned} \tag{12}$$

where the second equality is directly derived from a Schur's complement argument. Consider now any couple of points $(A_1, b_1, c_1), (A_2, b_2, c_2) \in epi_Q$ and any $\theta \in [0, 1]$. We clearly have that the convex combination

$$\begin{aligned} &\theta \begin{pmatrix} A_1 \otimes K^\dagger & b_1 \\ b_1^\top & c_1 \end{pmatrix} + (1-\theta) \begin{pmatrix} A_2 \otimes K^\dagger & b_2 \\ b_2^\top & c_2 \end{pmatrix} \\ &\qquad = \begin{pmatrix} \theta A_1 \otimes K^\dagger + (1-\theta) A_2 \otimes K^\dagger & \theta b_1 + (1-\theta) b_2 \\ \theta b_1^\top + (1-\theta) b_2^\top & \theta c_1 + (1-\theta) c_2 \end{pmatrix} \end{aligned} \tag{13}$$

still belongs to S_+^{nT+1}, which implies that

$$(\theta A_1 + (1-\theta) A_2, \theta b_1 + (1-\theta) b_2, \theta c_1 + (1-\theta) c_2) \in epi_Q \tag{14}$$

therefore proving that Q is jointly convex in b and A.

□

6.2 Cluster Multi-task Learning

We briefly recall here the Convex Multi-task Cluster Learning proposed in [13] and show that it can be cast in the same framework as that of our Sparse Kernel MTL model. In particular we comment what choice of constraint set $\mathcal{A}$ can be imposed on the structure matrix A to recover clustered structures of tasks.

In the setting proposed by [13], tasks are assumed to belong to one of r of unknown clusters, with r fixed a priori. While the original formulation is for the linear kernel, it can be easily extended to the non-linear setting of RKHSvv. Let $E \in \{0,1\}^{T\times r}$ be the binary matrix whose entry E_{st} has value 1 whenever a task s belongs to cluster t, and 0 otherwise. Let L be the normalized Laplacian of the Graph defined by E. Set $M = I - L$, and $U = \frac{1}{T}11^\top$. As we have observed in Eq. (6), the regularizer $\|f\|_{\mathcal{H}}$ depends on A^{-1}. The role of this term could be shaped to reflect the structure of the clusters encoded in the Laplacian L, hence in the matrix M. As noted in [13] $A^{-1}(M)$ can be chosen so that:

$$A^{-1}(M) = \epsilon_M U + \epsilon_B (M - U) + \epsilon_W (I - M), \tag{15}$$

where the first term is a global penalty on the average predictor, the second term penalizes the between cluster variance, and the third term penalizes the within cluster variance. Since M belongs to a discrete set, the authors propose a relaxation for M by constraining it to be in a convex set $\mathcal{S}_c = \{M \in S_+^T, 0 \preceq M \preceq I, \mathrm{tr}(M) = r\}$ which directly induces a set $\mathcal{A}$ of spectral constraints for A.